Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Symbolic Computation 40 (2005) 1053–1075
Journal of Symbolic Computation
www.elsevier.com/locate/jsc

Conquering inseparability: Primary decomposition and multivariate factorization over algebraic function fields of positive characteristic

Allan Steel*

School of Mathematics and Statistics F07, University of Sydney, NSW 2006, Australia

Received 4 November 2003; accepted 14 March 2005
Available online 24 June 2005

Abstract

Algebraic function fields of positive characteristic are non-perfect fields, and many standard algorithms for solving some fundamental problems in commutative algebra simply do not work over these fields. This paper presents practical algorithms for the first time for (1) computing the primary decomposition of ideals of polynomial rings defined over such fields and (2) factoring arbitrary multivariate polynomials over such fields. Difficulties involving inseparability and the situation where the transcendence degree is greater than one are completely overcome, while the algorithms avoid explicit construction of any extension of the input base field. As a corollary, the problem of computing the primary decomposition of a positive-dimensional ideal over a finite field is also solved. The algorithms perform very effectively in an implementation within the MAGMA Computer Algebra System, and an analysis of their practical performance is given.

Keywords: Algebraic function field; Non-perfect field; Inseparability; Primary decomposition; Polynomial factorization; Gröbner basis

* Tel.: +61 293 515 776; fax: +61 293 514 534.
E-mail address: allan@maths.usyd.edu.au.

0747-7171/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsc.2005.03.002

1. Introduction

Let K be a field of positive characteristic p which is finitely generated over its prime field. K can be written as

$$\mathbf{F}_q(t_1, \ldots, t_k)[\alpha_1, \ldots, \alpha_r]/I,$$

for transcendental generators t_i and algebraic generators α_j, where I is a maximal ideal of the polynomial ring $\mathbf{F}_q(t_1, \ldots, t_k)[\alpha_1, \ldots, \alpha_r]$ and q is a power of p. This paper presents practical algorithms for the first time for (1) computing **primary decomposition** of ideals of polynomials rings defined over K and (2) **factoring** multivariate polynomials over K. The algorithms work for arbitrary p, thus covering the important situation where p is small where there can be many practical difficulties involving inseparable field extensions.

Consider first the computation of the primary decomposition of an ideal. For a positive-dimensional ideal of a polynomial ring defined over any field K, there is a standard reduction to one or more computations of the primary decomposition of related zero-dimensional ideals over a transcendental extension of K (see Section 5.3 below).

So the challenge is the computation of the primary decomposition of a zero-dimensional ideal defined over a function field. A good complete algorithm which handles this, in the case where the function field is perfect, is given by Gianni et al. (1988, Section 7) (and effectively in Becker and Weispfenning (1993, Chapter 8)), but this algorithm does not work when the function field has positive characteristic. Instead, Gianni et al. (1988, Section 6) gives an algorithm for a zero-dimensional ideal over any field K, assuming one can factor polynomials over an algebraic extension of K. However, the algorithm is quite complicated, involving recursive calls on certain ideal components which are constructed for each variable, and the algebraic field extensions are constructed as quotients by multivariate maximal ideals, which will not yield simple algebraic extensions in general, and may be very non-trivial. The authors refer to methods given by Davenport and Trager (1981) to perform the required factorizations.

Now the paper Davenport and Trager (1981) presents methods for factoring a polynomial over a field which is finitely generated over its prime field. However, for the case of algebraic function fields of positive characteristic, there are inadequacies:

1. The algorithm assumes that the input polynomial is squarefree, but no algorithm is given for computing the squarefree part of a polynomial in this situation.
2. Furthermore, it is insufficient to pass a squarefree polynomial to Trager's norm-based algorithm `sqfr-norm` of Trager (1976); the polynomial must be **separable** to enable a squarefree norm to be found (see Theorem 6.2 below).
3. If the transcendence degree of the input field is greater than one, then the field may not be isomorphic to a simple algebraic extension over the base transcendental field. So the norm-based algorithm will fail to reduce the problem to one algebraic generator in such cases. The authors do refer to a technique for getting around purely inseparable extensions, but there is no complete algorithm given for handling this in the context of several algebraic extensions.

So to the best of the present author's knowledge, there has been no effective complete algorithm given for the factorization of a multivariate polynomial over a non-perfect algebraic

function field, and there has thus also been no effective complete algorithm given for the primary decomposition of an ideal over such a field. In any case, there has hitherto been no implementation available of any algorithms for solving these problems in full generality.

The approach taken in this paper for solving these problems is simpler: we in fact solve the primary decomposition problem first (Section 5), and this leads to a solution of the factorization problem (Section 6). Our general approach thus follows the opposite order to the one proposed above, and has the following advantages:

1. In the algorithms presented here, the only extension fields which need to be considered are *purely inseparable extensions* (see Section 2.1) of the input field. Furthermore, as explained in Section 2.3 below, one never needs to construct these extensions explicitly: they are easily simulated in practice by computations involving the original input field only. So the algorithms certainly do not need to factor over any non-trivial extension of the input field.
2. The main algorithm for computing the primary decomposition of a zero-dimensional ideal uses techniques which are similar to those used by Kemper (2002) for computing the radical of a zero-dimensional ideal: we 'virtually' extend the base field by only purely inseparable extensions until all the relevant polynomials become separable. Then suitable modifications of standard algorithms for perfect fields can be applied, so our algorithms involve only natural extensions of already existing techniques. At the end of each algorithm, we move back to the original field using a Gröbner basis elimination technique (but we are able to avoid using the Buchberger algorithm).
3. The algorithm for factoring involves an easy application of a few key components of the primary decomposition algorithm plus standard code for factorization over perfect fields, so does not need much extra code to implement. Also, the factorization algorithm works directly for multivariate inputs, instead of being only a univariate algorithm on top of which one would have to develop another non-trivial multivariate evaluation/lifting factorization algorithm.
4. In all of this, we easily cover fields which can have **both** an arbitrary number of transcendental generators and an arbitrary number of algebraic generators, without special handling.

The paper is organized as follows. Sections 2–4 develop the basic theory and subalgorithms needed for our approach. Then Sections 5 and 6 give the main algorithms for computing primary decompositions and factoring polynomials, respectively. Finally, Section 7 analyzes the performance of the algorithm in practice.

All of the algorithms of this paper have been implemented in the MAGMA computer algebra system (Bosma et al., 1997), and have been available in version V2.10 since April 2003.

2. Field extensions

2.1. Purely inseparable extensions

We first note the basic properties of the kinds of field extensions which will arise commonly throughout the paper.

Definition 2.1. Suppose K is a field of characteristic p. A extension field L of K is called a **purely inseparable extension (PIE)** of K if for all $x \in L$, $x^{p^k} \in K$ for some integer $k \geq 0$.

Lemma 2.2. *Suppose L is a PIE of K of characteristic p.*

(a) *For any $g \in L[x_1, \ldots, x_n]$, there exists an integer k with $k \geq 0$ such that $g^{p^k} \in K[x_1, \ldots, x_n]$.*
(b) *If $g \in L[x_1, \ldots, x_n]$ is irreducible over L, and k is minimal such that $f = g^{p^k} \in K[x_1, \ldots, x_n]$, then f is irreducible over K.*
(c) *If $f \in K[x_1, \ldots, x_n]$ is irreducible over K, then $f = c \cdot g^{p^k}$ for some $g \in L[x_1, \ldots, x_n]$, $k \geq 0$, and $c \in K$, where g is irreducible over L.*
(d) *If $f \in K[x_1, \ldots, x_n]$ and $f = c \cdot \prod_{i=1}^{k} g_i{}^{e_i}$ is a factorization of f over L into powers of irreducibles (with the scalar $c \in L$), then $g_i{}^{e_i} \in K[x_1, \ldots, x_n]$.*

Proof.

(a) Since the coefficients of g are in L, there exists some k with $k \geq 0$ such that $x^{p^k} \in K$ for every coefficient x of g. But g^{p^k} has these coefficients (with the monomial exponents also scaled), so is in $K[x_1, \ldots, x_n]$.
(b) If $f = f_1 \cdot f_2$ with $f_1, f_2 \in K[x_1, \ldots, x_n]$ coprime and non-constant, then $g^{p^k} = f_1 \cdot f_2$ with f_1 and f_2 still coprime over L, which contradicts the irreducibility of g over L. So $f = c \cdot f_1{}^e$ with $f_1 \in K[x_1, \ldots, x_n]$ irreducible over K and $c \in K$. But since g^{p^k} is the factorization of $f = c \cdot f_1{}^e$ over L, e must be a power of p. By the minimality of k, we must have $e = 1$, so f is irreducible over K.
(c) Let $g \in L[x_1, \ldots, x_n]$ be an irreducible factor of f over L and let k be minimal with $h = g^{p^k} \in K[x_1, \ldots, x_n]$. By (b), h is irreducible over K. Since g divides f and h, g divides their GCD, but this cannot be one (because g is irreducible), so since h is irreducible, we have $f = c \cdot h$ for some $c \in K$.
(d) Let $f = b \cdot \prod_{i=1}^{m} h_i{}^{s_i}$ be a factorization of f over K into powers of irreducibles. By (c), $h_i = d_i^{(p^{k_i})}$ where $d_i \in L[x_1, \ldots, x_n]$ is irreducible over L and $k_i \geq 0$ for each i. So $f = b \cdot \prod_{i=1}^{m} d_i{}^{(p^{k_i} s_i)}$ is a factorization of f over L into powers of irreducibles. Then we must have $m = k$ and with suitable renumbering and rescaling, we can write $d_i = g_i$, $e_i = p^{k_i} s_i$. Then $g_i^{e_i} = d_i^{(p^{k_i} s_i)} = h_i^{s_i} \in K[x_1, \ldots, x_n]$. □

2.2. Intersecting an ideal with a polynomial ring over a subfield

Our general approach will be to extend the input field K to a PIE L over which we can solve our problems. At the end of each main algorithm, we have to move back to the original field K. The following algorithm works for general algebraic extensions, but we will only apply it to purely inseparable extensions in this paper.

Algorithm IDEALOVERSUBFIELD(K, I)
INPUT:

1. A field K.

2. An algebraic extension L of K, of the form

$$L = K[\alpha_1, \ldots, \alpha_r]/\langle s_1, \ldots, s_l \rangle,$$

where the set of the s_i forms a reduced Gröbner basis of the defining ideal with respect to a monomial order $<_\alpha$.
3. An ideal I (not necessarily zero dimensional) of $L[x_1, \ldots, x_n]$ with Gröbner basis B with respect to a monomial order $<_x$.

OUTPUT: $I \cap K[x_1, \ldots, x_n]$.
STEPS:

1. Let ϕ be the natural epimorphism:

$$K[y_1, \ldots, y_n, \beta_1, \ldots, \beta_r] \to (K[\alpha_1, \ldots, \alpha_r]/\langle s_1, \ldots, s_l \rangle)[x_1, \ldots, x_n],$$

where $y_i \mapsto x_i$, and $\beta_i \mapsto \alpha_i$. Form the ideal J of $K[y_1, \ldots, y_n, \beta_1, \ldots, \beta_r]$ with basis $B_1 = \{B', t_1, \ldots, t_l\}$, where B' is a set of preimages of the elements of B under ϕ, and t_i is a preimage of s_i under ϕ, for each i.
2. Note that under the block order naturally derived from the given orders ($<_x$ for the y_i and $<_\alpha$ for the β_j) and with $y_i > \beta_j$, B_1 is a Gröbner basis of J. Use a Gröbner basis order change algorithm such as the Gröbner Walk to compute the reduced Gröbner basis B_2 of J with respect to the block order derived from the given orders and with $\beta_j > y_i$.
3. Let E be the set of polynomials in B_2 in which no β_j occurs with non-zero degree. Return the ideal of $K[x_1, \ldots, x_n]$ generated by $\phi(E)$.

Theorem 2.3. *Algorithm* IDEALOVERSUBFIELD *is correct.*

Proof. This is simply a standard Gröbner basis elimination technique. The ideal J in Step 1 is clearly the inverse image of I under ϕ. Now

$$\begin{aligned} J \cap K[y_1, \ldots, y_n] &= \phi^{-1}(I) \cap \phi^{-1}(K[x_1, \ldots, x_n]) \\ &= \phi^{-1}(I \cap K[x_1, \ldots, x_n]), \end{aligned}$$

and Step 3 returns

$$\phi(J \cap K[y_1, \ldots, y_n]) = I \cap K[x_1, \ldots, x_n]. \quad \square$$

Remark 2.4. For details on the Gröbner Walk algorithm mentioned in Step 2, see Amrhein et al. (1996) and Collart et al. (1997) and further discussion concerning its performance in Section 7.3 below. The great advantage of using the basis change algorithm is that IDEALOVERSUBFIELD does not need to call the Buchberger algorithm at any point (under the assumption that a Gröbner basis is given for the input ideal).

Example 2.5. To see how IDEALOVERSUBFIELD works, let $K = \mathbf{F}_2(t, u)$, $L = \mathbf{F}_2(\sqrt[4]{t}, \sqrt{u})$, and let

$$I = \langle x + y + \sqrt[4]{t}, y^2 + \sqrt[4]{t}y + \sqrt{u} \rangle \lhd L[x, y].$$

Write $T = \sqrt[4]{t}$, $U = \sqrt{u}$ (corresponding to α_1, α_2 in the algorithm). Then,

$$B_1 = \left\{ \begin{array}{l} x + y + T, \\ y^2 + Ty + U, \\ T^4 + t, \\ U^2 + u \end{array} \right\}$$

is clearly a Gröbner basis of J with the monomial order $x > y > T > U$. When we change the monomial order of J to $T > U > x > y$, we obtain the Gröbner basis:

$$B_2 = \left\{ \begin{array}{l} T + x + y, \\ U + xy, \\ x^2 + \frac{1}{u}y^6 + \frac{t}{u}y^2, \\ y^8 + ty^4 + u^2 \end{array} \right\}.$$

So the last two polynomials generate $I \cap K[x, y]$. Note that the polynomial for y in B_2 is the fourth power of the second original polynomial, which is not surprising, but the polynomial involving x and y in B_2 cannot easily be derived by inspection from the original ideal.

2.3. Simulating purely inseparable extensions

Suppose K is a field of characteristic p, q is a power of p, and $k \geq 1$. In the algorithms described below, we will sometimes need to construct a PIE field L of K such that L contains p^k-th roots of elements of K. We describe a trick which avoids the need to construct L as an explicit extension, for the two types of field K which we will encounter.

(1) Suppose $K = \mathbf{F}_q(t_1, \ldots, t_k)$. Define the embedding $\phi : K \to K$ by $t \mapsto t^{p^k}$ and let $S = \phi(K)$. Then clearly K is isomorphic to S under ϕ, while K also contains the p^k-th root of any element of S.
(2) Suppose $K = \mathbf{F}_q(t_1, \ldots, t_k)[\alpha]/\langle f \rangle$ where $f \in \mathbf{F}_q(t_1, \ldots, t_k)[\alpha]$ is irreducible, so K is a field. Define $\phi : K \to K$ by $g + \langle f \rangle \mapsto g^{p^k} + \langle f \rangle$. It is elementary to check that ϕ is a well-defined homomorphism. As ϕ is a non-zero homomorphism from a field to a field, ϕ is injective, so again letting $S = \phi(K)$, we have that K is isomorphic to S under ϕ, while K clearly also contains the p^k-th root of any element of S.

In either case, we can conceptually map the elements of K under ϕ, and then proceed under the perspective that S represents the original K while K now represents the desired extension field L having p^k-th roots of elements of the original K.

Applying this in the implementation, if we have a polynomial h with coefficients in K, we multiply the exponents of the t_i indeterminates (and α, if present) by p^k and continue working with these new coefficients of h (which now possess p^k-th roots) explicitly represented in K. But we also keep a **level** k associated with h, indicating that the exponents have been scaled by p^k. If we construct an ideal generated by polynomials having different levels, then it is trivial to do further rescaling on the exponents so that all the polynomials in the ideal have the one common level (the maximum of the input levels).

When the above elimination algorithm IDEALOVERSUBFIELD (Section 2.2) is called on an ideal having polynomials of common level k, we rewrite the coefficients in terms

of new u_i generators (and a new β generator, if needed), include relations expressing the original t_i generators (and α, if present) as p^k-th powers of the new generators, and then we can compute the result over the original K.

Thus although the algorithms in this paper are presented, for simplicity of exposition, as constructing purely inseparable extension fields as needed, we can always compute in practice with the original base field alone and never need to extend it explicitly.

3. Separable polynomials

In this section, let K be a field of characteristic p, such that one can effectively compute p^k-th roots of elements of K.

Definition 3.1. We call a polynomial $f \in K[x_1, \ldots, x_n]$ **separable** if f is squarefree over K and over any extension field of K.

Lemma 3.2. *Suppose* $f \in K[x_1, \ldots, x_n]$. *If* $\mathrm{GCD}(f, \frac{\partial f}{\partial x_v}) = 1$, *for some* v *with* $1 \leq v \leq n$, *then* f *is separable.*

Proof. If f is inseparable, then $f = g^e h$ for some $g, h \in \bar{K}[x_1, \ldots, x_n]$, for some algebraic closure $\bar{K}$ of K and with $g \neq 1$ and $e > 1$; then for any v, $\mathrm{GCD}(f, \frac{\partial f}{\partial x_v}) \neq 1$, since g divides the GCD (whether p divides e or not is irrelevant). □

The converse is also true in the case of one variable (Becker and Weispfenning, 1993, Proposition 7.33), but false in the case of two or more variables. For example, $(x^2 + y)(x + y^2) \in \mathbf{F}_2[x, y]$ is separable, but the GCDs with the derivatives are both non-trivial.

We now present an algorithm for computing the **separable factorization** of an arbitrary multivariate polynomial f. This is similar to the squarefree factorization of f, except that all the factors in the result are **separable** (not just squarefree), and may lie over an extension of the input field.

Algorithm SEPARABLEFACTORIZATION(f)
INPUT: $f \in K[x_1, \ldots, x_n]$, where K is a field of characteristic p, such that one can effectively compute p^k-th roots of elements of K.
OUTPUT: A sequence S containing pairs of the form (g_i, e_i) such that the $g_i \in L[x_1, \ldots, x_n]$ (for some PIE L of K) are separable, and f equals the product of the $g_i^{e_i}$ (times a constant in L).
STEPS:

1. Subalgorithm SF(f, s)
   {
   If f is constant then return the empty sequence.
   Let k be maximal such that p^k divides the x_i-exponents
   (for all $i = 1, \ldots, n$), of all monomials of f.
   If $k > 0$
   {
   Extend the base field of f as necessary so that all

coefficients of f have p^k-th roots.
Set $f :=$ the p^k-th root of f.
Set $s := s \cdot p^k$.
}
Let v be such that $\frac{\partial f}{\partial x_v} \neq 0$.
Set $g := \mathrm{GCD}(f, \frac{\partial f}{\partial x_v})$.
If $g = 1$ then Return $[(f, s)]$.
Return $\mathtt{SF}(g, s)$ concatenated with $\mathtt{SF}(f/g, s)$.
}

2. Set $S := \mathtt{SF}(f, 1)$.
Find covering field L and lift all factors in S to be over L.
Collect equal factors in S, combining multiplicities.
Return S.

Theorem 3.3. *Algorithm* SEPARABLEFACTORIZATION *is correct.*

Proof. We need only show that the subalgorithm SF satisfies the output condition, except that there may possibly be repetitions of the separable polynomials g_i; the outer statements in Step 2 simply collect these, combining the multiplicities, and move to a common extension field.

So consider SF. The constant handling is trivially correct. If $k > 0$, then clearly one can compute the p^k-root of f by dividing all exponents by p^k and taking p^k-th roots of the coefficients, once the base field has been extended accordingly; the multiplicity scale s is also correctly updated. After the If-statement, there must exist a v such that p does not divide all the exponents in x_v of f, so $\frac{\partial f}{\partial x_v} \neq 0$, and the GCD g cannot equal f. If $g = 1$, then f must be separable by Lemma 3.2, so the Return-statement is correct. Otherwise the returned result correctly contains separable polynomials by induction, and will give a separable factorization of f. The algorithm terminates because g and f/g must both have strictly smaller degree than f in the variable x_v. □

Remark 3.4.

(1) We have given a recursive algorithm here for simplicity of presentation, but in practice we use an iterative algorithm to remove repeated factors whose multiplicity is not a power of p (one can use a modification of standard squarefree factorization algorithms).
(2) This algorithm will only be applied below to the two types of field discussed in Section 2.3. Following the techniques outlined there, in the implementation we only compute in the original field K in practice, and each polynomial in the output of SF has a level associated with it. Finding the covering field simply means moving all the polynomials to the maximum level, so they all have a common level.
(3) This algorithm is similar in spirit to Kemper's algorithm (Kemper, 2002, Alg. 1) for computing the separable part of polynomial.

Example 3.5. Suppose $K = \mathbf{F}_2(t, u)$ and $R = K[x, y]$.

(1) Let $f = x^2y^2 + t \in R$. Then since 2 divides all monomial exponents of f, the base field is extended to $\mathbf{F}_2(\sqrt{t}, u)$ and f is replaced with $f_1 = xy + \sqrt{t}$. Then $\mathrm{GCD}(f_1, \frac{\partial f_1}{\partial x}) = 1$, so f_1 is separable and the result is $[(f_1, 2)]$.

(2) Let $f = (x^4+t)(x^2+y+u^2) = x^6+x^4y+u^2x^4+tx^2+ty+tu^2 \in R$. The If-statement is skipped, since 2 does not divide all y-exponents of f. Then $d = \frac{\partial f}{\partial y} = x^4+t$ is non-zero, and $g = \mathrm{GCD}(f, d) = d$. Recursing on g, we see that 4 divides all exponents, and the inner result is $[(x + \sqrt[4]{t}, 4)]$. When we recurse on $f_2 = f/g = x^2 + y + u^2$, 2 does not divide the y-exponent and $\mathrm{GCD}(f_2, \frac{\partial f_2}{\partial y}) = 1$, so f_2 is separable. So the final result is $[(x+\sqrt[4]{t}, 4), (x^2+y+u^2, 1)]$ with the polynomials lying in $\mathbf{F}_2(\sqrt[4]{t}, u)[x, y]$.

4. Shape bases and separable ideals

Definition 4.1. Let $R = K[x_1, \ldots, x_n]$, for a field K, and suppose I is a **radical** zero-dimensional ideal of R. We call $s = \sum_{i=1}^{n} c_i x_i$, with $c_i \in K$, a **shape basis generator** of I if $(s + I)$ generates the quotient R/I (as an algebra over K).

If s is a shape basis generator of I, it is well known that by introducing a new variable z, and letting I' be the ideal of $R' = K[x_1, \ldots, x_n, z]$ generated by the embedding of I in R' and $(z - s)$, then the minimal reduced Gröbner basis of I' with respect to the lexicographical order with $x_1 > \cdots > x_n > z$ has the form

$$\{x_1 - f_1(z), \ldots, x_n - f_n(z), g(z)\},$$

where the f_i and g are univariate polynomials in $K[z]$. Such a basis is called a **shape basis**.

If I is also maximal, then I has a shape basis generator if and only the field R/I has a primitive element over K (i.e., if and only if R/I is a simple algebraic extension of K).

To solve the main kinds of problems discussed in this paper, the standard algorithms usually need to compute shape basis generators. Now Becker and Weispfenning (1993, Section 8.6) effectively shows that over a **perfect** infinite field, a zero-dimensional radical ideal always possesses a shape basis generator. But over a non-perfect infinite field, this may not be the case. As an example, let $K = \mathbf{F}_2(t, u)$, $R = K[\alpha, \beta]$, and let I be the radical zero-dimensional ideal $\langle \alpha^2 + t, \beta^2 + u \rangle$ of R. Now $L = R/I$ is a degree-4 field extension of K, with algebra basis $[1, \alpha, \beta, \alpha\beta]$ over K, but L does not have a primitive element over K, since the square of any element of L clearly lies in K, so the minimal polynomial of any element of L has degree at most 2. Thus I does not possess a shape basis generator, despite being radical.

We address this problem by introducing the concept of a **separable** radical. Such a radical will always possess a shape basis generator.

First note some terminology: if I is any ideal of $K[x_1, \ldots, x_n]$, and L is an extension field of K, then "I over L" will refer of course to the ideal of $L[x_1, \ldots, x_n]$ generated by the embedding of I in $L[x_1, \ldots, x_n]$.

Definition 4.2. Suppose I is a zero-dimensional ideal of $K[x_1, \ldots, x_n]$, for a field K. We say that I is **separable** if I is radical over K and remains radical over any extension field

of K. Naturally, we can also call I a **separable radical** of J if I is the radical of J over the field of definition of I, and I is also separable.

Lemma 4.3 (*Becker and Weispfenning, 1993, Lem. 8.13* (SEIDENBERG'S LEMMA 92)). *Suppose I is a zero-dimensional ideal of $K[x_1, \ldots, x_n]$, for a field K. If $I \cap K[x_i]$ contains a separable polynomial for $i = 1, \ldots, n$, then I is radical over K.*

Corollary 4.4. *Suppose I is a zero-dimensional ideal of $K[x_1, \ldots, x_n]$, for a field K. If I satisfies the condition of Lemma 4.3, then I is separable.*

Over a non-perfect field, a radical ideal is not necessarily separable, but over a perfect field, the concepts coincide, of course. We can now present the critical theorem which will ensure the termination of the main algorithms.

Theorem 4.5. *Let I be a separable zero-dimensional ideal of $K[x_1, \ldots, x_n]$ where K is an infinite field. Then I possesses a shape basis generator which can be computed effectively.*

Proof. As defined in Becker and Weispfenning (1993, Def. 8.67), a zero-dimensional ideal I of $K[x_1, \ldots, x_n]$ is said to be in normal position with respect to some variable x_i if the i-th coordinates of the points of the affine variety of I over an algebraic closure of K are distinct.

We introduce a new variable z, and work in $R' = K[x_1, \ldots, x_n, z]$. Since K is infinite and I is radical over K, Becker and Weispfenning (1993, Lem. 8.76) shows that there exists a finite subset C of K^n, effectively computable from I, such that for some $c = (c_1, \ldots, c_n) \in C$, the extended ideal of R' generated by the embedding of I and $(z - \sum_{i=1}^{n} c_i x_i)$ is in normal position with respect to z. With a suitable substitution, we may thus consider I to be in normal position.

Now since I is separable, I equals the radical of I over any extension field of K. The proof of Becker and Weispfenning (1993, Theorem 8.32) shows that in this very situation, the dimension of the vector space $K[x_1, \ldots, x_n]/I$ equals the number of points in the affine variety of I. Then the proof of Becker and Weispfenning (1993, Proposition 8.77) shows that because of this equality, and because I is in normal position, I possesses a shape basis (and a shape basis generator is computed explicitly from c above). □

Corollary 4.6. *The field $K[x_1, \ldots, x_n]/I$ (for any infinite field K and maximal ideal I) has an effectively computable primitive element over K if I is separable.*

Remark 4.7. In this paper, in practice, we will have a separable ideal I of $K[x_1, \ldots, x_n]$, where K is a function field of small characteristic with at least one transcendental generator (otherwise the algorithms in this paper would not be needed). Then to compute a shape basis generator of I we repeatedly let s be $\sum_{i=1}^{n} c_i x_i$, where each c_i is constructed from a random small-degree polynomial in the transcendental generators of K, until the minimal polynomial of s in $K[x_1, \ldots, x_n]/I$ has the maximal degree. We use a variant of the FGLM algorithm (Faugère et al., 1993) to compute the minimal polynomial, and this also gives the shape basis explicitly, when the maximal degree is attained. Note that this is of course similar to what one does in characteristic zero; the point of the Theorem is that this algorithm always terminates when given a separable ideal!

5. Primary decomposition

In this section we solve the problem of computing the primary decomposition of an ideal defined over a general algebraic function field. The first subsection presents the principal algorithm which handles a zero-dimensional ideal over a rational function field. Then two following subsections use simple applications of this algorithm to handle more general ideals. See Becker and Weispfenning (1993, Chapter 8) for basic background on primary decomposition of ideals.

5.1. Zero-dimensional ideals over rational function fields

In this subsection we present the main algorithm for computing the primary decomposition of a zero-dimensional ideal defined over a rational function field.

The algorithm has two main steps. In the first step, we compute a partial decomposition into components over appropriate purely inseparable extension fields, where the radical of each component is separable. In the second step, we compute shape bases of these radicals and thereby can fully decompose each component, and finally we move the results to the corresponding ideals over the original field.

Algorithm PRIMARYDECOMPOSITION(I)
INPUT: I, a zero-dimensional ideal of $K[x_1, \ldots, x_n]$, where K is the rational function field $\mathbf{F}_q(t_1, \ldots, t_k)$, with q a power of a prime p.
OUTPUT: The primary decomposition of I over K, returned as a set D of triples (Q_i, P_i, S_i), where Q_i is primary and the intersection of the Q_i is I, P_i is the corresponding prime for Q_i, S_i is the radical of P_i over some PIE L_i of K, and S_i is separable.
STEPS:

```
1.  Set R := {(I, I, 1)}.
    For v := 1 to n do
    {
        Set R' := {}.
        For each (Q, S, l) triple in R do
        {
            Set f := the monic generator of Q ∩ K[x_v].
            Set T := SEPARABLEFACTORIZATION(f).
            For each (g, e) in T do
            {
                Set L := the base field over which g is defined.
                For each irreducible factor h of g over L do
                {
                    Set Q' := Q + ⟨h^e⟩.
                    Set S' := S + ⟨h⟩.
                    Set l' := the smallest power of p which is ≥ e.
                    Insert (Q', S', Max(l, l')) into R'.
                }
```

```
            }
        }
        Set R := R'.
    }

2.  Set D := {}.
    For each (Q, S, l) triple in R do
    {
        Set s := a shape basis generator of S.
        Set L := the base field of S.
        Set f := the minimal polynomial of s in L[x_1, ..., x_n]/S.
        For each irreducible factor g of f do
        {
            Set t := g(s).
            Set Q' := Q + ⟨t^l⟩.
            Set S' := S + ⟨t⟩.
            Set P' := IdealOverSubfield(K, S').
            Insert (Q', P', S') into D.
        }
    }
    return D.
```

Theorem 5.1. *Algorithm* PRIMARYDECOMPOSITION *is correct.*

Proof. Step 1 is akin to algorithm PREDEC in Becker and Weispfenning (1993, Table 8.1). At the end of the step, R contains triples of the form (Q_i, S_i, l_i). For each i, the ideal S_i, defined over a PIE L_i of K, is the radical of Q_i over L_i and is separable by Corollary 4.4. By Becker and Weispfenning (1993, Lem 8.5), I equals the intersection of all the Q_i.

Also, each S_i is generated over L_i by Q_i and polynomials $h_{i,j}$ (for $j = 1, \ldots, m_i$, say). By Lemma 2.2(d), we have each $h_{i,j}^{e_{i,j}} \in K[x_1, \ldots, x_n]$, so Q_i is an ideal of $K[x_1, \ldots, x_n]$. Now $l_i \geq e_{i,j}$ for each j, so $h_{i,j}^{l_i}$ is in Q_i generated over L_i, but since l_i is also a power of p and at least as large as $e_{i,j}$, the multiplicity of p in l_i is at least the multiplicity in $e_{i,j}$, so $h_{i,j}^{l_i}$ is also in $K[x_1, \ldots, x_n]$ and thus $h_{i,j}^{l_i}$ is in Q_i generated over K. Since SEPARABLEFACTORIZATION returns the minimal PIE of K necessary to define each $h_{i,j}$, we also have $a^{l_i} \in K, \forall a \in L_i$. So for any $s \in S_i$, applying the Frobenius homomorphism, we also have $s^{l_i} \in Q_i$ (generated over K).

Step 2 computes the full decomposition of the current (Q_i, S_i, l_i) components and the corresponding primes over the original field K, returning triples of the form (Q'_j, P'_j, S'_j), where each $S'_j \lhd L'_j[x_1, \ldots, x_n]$ and is separable. Because each S_i is a separable radical, it possesses a shape basis generator s by Theorem 4.5. By the discussion on l_i above, we again have $Q'_j \lhd K[x_1, \ldots, x_n]$, and from the factorization of f, each Q'_j over L'_j is primary over L'_j with associated prime S'_j over L'_j (see Becker and Weispfenning (1993, Proposition 8.69) for the basic situation). Also, Q'_j is primary over K: if $a, b \in$

$K[x_1, \ldots, x_n]$ and $ab \in Q'_j$ but $a \notin Q'_j$, then $a \notin Q'_j$ over L'_j, so $b^\nu \in Q'_j$ over L'_j for some $\nu \geq 1$, but then $b^\nu \in Q'_j$ also.

Thus at the end of Step 2, the intersection of the Q'_j gives the primary decomposition of I over K. By Theorem 2.3, $P'_j = S'_j \cap K[x_1, \ldots, x_n]$. Now $Q'_j \lhd K[x_1, \ldots, x_n]$ and $Q'_j \subseteq S'_j$, so $Q'_j \subseteq P'_j$. If $f^e \in Q'_j$ for some $f \in K[x_1, \ldots, x_n]$ and $e \geq 1$, then clearly $f \in S'_j \cap K[x_1, \ldots, x_n] = P'_j$, so P'_j is the radical (corresponding prime) of Q'_j over K. □

Remark 5.2.

(1) The base field K is of the type covered in point (1) of Section 2.3. So as pointed out there and in Remark 3.4(2), we do not need to extend the base field K in practice, but each separable g arising in Step 1 has a level associated with it, indicating how the exponents of its coefficients (rational functions) have been scaled. We only need keep track of one level for each Q_i and use the same level for its corresponding S_i; each final call to IDEALOVERSUBFIELD then takes the corresponding level into account. Also, the factorization of each g over its field L in Step 1 is also done in practice over K, and this is valid because of the underlying isomorphism. Factorization over K is well covered by existing algorithms (see Section 7.4 for details).

(2) The univariate generator of the elimination ideal $Q \cap K[x_v]$ could be computed each time in Step 1 via the FGLM algorithm (Faugère et al., 1993), but the Gröbner Walk has been found to be very effective for moving successively to the Gröbner basis of each ideal with respect to a block 'univariate elimination' order for each x_v (where x_v is considered less than all the other variables). We keep the current Gröbner basis of each ideal at every stage, so the next univariate elimination ideal can be 'walked to' each time. So it is easy to supply Gröbner bases for the calls to IDEALOVERSUBFIELD, and we never need to use the Buchberger algorithm, assuming that we start with a Gröbner basis for the input ideal I.

We also present here a simpler algorithm SEPARABLERADICAL for computing a separable radical of an ideal over an appropriate extension field, since this will be needed in the next section. The algorithm is essentially equivalent to Kemper's algorithm for computing the radical (Kemper, 2002, Alg. 6), without the final restoration to the original field, and involves a simpler combination of a few of the components of PRIMARYDECOMPOSITION. There is only one separable radical S occurring, which is constructed by taking the ideal generated by I and the separable parts of the univariate elimination ideal generators of I (with no factoring needed). We give a formal description of the algorithm for completeness.

Algorithm SEPARABLERADICAL(I)
INPUT: I, a zero-dimensional ideal of $K[x_1, \ldots, x_n]$, where $K = \mathbf{F}_q(t_1, \ldots, t_k)$.
OUTPUT: An ideal S of $L[x_1, \ldots, x_n]$, where L is some PIE of K, such that S is the radical of I over L and S is separable.

STEPS:

Set $S := I$.
For $v := 1$ to n do
{
  Set $f :=$ the monic generator of $I \cap K[x_v]$.
  Set $T :=$ SEPARABLEFACTORIZATION(f).
  Set $g :=$ the product of the polynomials in T (ignoring multiplicities).
  Set $S := S + \langle g \rangle$.
}
Return S.

Note that one can then call IDEALOVERSUBFIELD(K, S) to compute the radical of I over K, as Kemper effectively does.

Example 5.3.

(1) Let $K = \mathbf{F}_2(t)$, and let

$$I = \langle x^2 + y, (y^2 + t^2)(y^2 + t) \rangle \lhd K[x, y].$$

Step 1 of PRIMARYDECOMPOSITION yields two components:

$$Q_1 = \langle x^2 + y, y^2 + t^2 \rangle, \quad S_1 = \langle x + \sqrt{t}, y + t \rangle, \quad l_1 = 2,$$

and

$$Q_2 = \langle x^2 + y, y^2 + t \rangle, \quad S_2 = \langle x + \sqrt[4]{t}, y + \sqrt{t} \rangle, \quad l_2 = 4.$$

Both S_1 are S_2 are already in shape basis form, and are easily seen to be prime over $\mathbf{F}_2(\sqrt[4]{t})$. After calling IDEALOVERSUBFIELD on K and each of these to obtain P_1 and P_2, respectively, we finish with

$$Q_1 = \langle x^2 + y, y^2 + t^2 \rangle, \quad P_1 = \langle x^2 + y, y + t \rangle.$$

and

$$Q_2 = \langle x^2 + y, y^2 + t \rangle, \quad P_2 = \langle x^2 + y, y^2 + t \rangle.$$

Note that $Q_1 \neq P_1 \neq S_1$, while $Q_2 = P_2 \neq S_2$. So neither of the P_i is separable (despite being radical over K).

(2) Let $K = \mathbf{F}_2(t, u)$, and let

$$I = \langle \alpha^2 + t\beta, \beta^4 + \beta^2 + \gamma, \gamma^2 + tu \rangle \lhd K[\alpha, \beta, \gamma].$$

Step 1 of PRIMARYDECOMPOSITION finds only one component, and the separable radical is

$$S = \langle \alpha + \sqrt{t}\beta + \sqrt[8]{t^5}\sqrt[8]{u}, \ \beta^2 + \beta + \sqrt[4]{t}\sqrt[4]{u}, \ \gamma + \sqrt{t}\sqrt{u} \rangle.$$

(SEPARABLERADICAL would return this same S.) The corresponding prime over the original field, equal to $S \cap K[\alpha, \beta, \gamma]$, turns out to be I itself, so I is maximal, and $F_K = K[\alpha, \beta, \gamma]/I$ is a field. S can be written as

$$\langle \alpha + T^4\beta + T^5U, \ \beta^2 + \beta + T^2U^2, \ \gamma + T^4U^4 \rangle,$$

where $T = \sqrt[8]{t}, U = \sqrt[8]{u}$. S already has a shape basis in this form, with shape basis generator β. We will continue this example in Example 6.5(2), where we will factor a polynomial over the field F_K.

5.2. Zero-dimensional ideals over algebraic function fields

Let K be the rational function field $\mathbf{F}_q(t_1, \ldots, t_k)$ and suppose I is a maximal ideal of $K[\alpha_1, \ldots, \alpha_r]$, so $F_K = K[\alpha_1, \ldots, \alpha_r]/I$ is a finitely presented algebraic function field. A simple application of the previous algorithm allows us to compute the primary decomposition of a zero-dimensional ideal J of $F_K[x_1, \ldots, x_n]$.

Algorithm PRIMARYDECOMPOSITIONALGEBRAIC(J)
INPUT: A zero-dimensional ideal J of $F_K[x_1, \ldots, x_n]$, where F_K is as above.
OUTPUT: The primary decomposition of J.
STEPS:

1. Let ϕ be the natural epimorphism:

$$K[y_1, \ldots, y_n, \beta_1, \ldots, \beta_r] \to F_K[x_1, \ldots, x_n]$$

given by $y_i \mapsto x_i, \beta_j \mapsto \alpha_j$.
2. Let $J' = \phi^{-1}(J)$, generated by preimages of a basis of J and a basis for the kernel of ϕ (obtained by mapping α_j to β_j in a basis of I). This is very similar to Step 1 of IDEALOVERSUBFIELD. Note that J' is zero dimensional since J and I are.
3. Call PRIMARYDECOMPOSITION(J') to obtain the primary decomposition $(Q'_1, P'_1), \ldots, (Q'_s, P'_s)$ of J'.
4. Set $P_i = \phi(P'_i)$ and $Q_i = \phi(Q'_i)$ for each i, and return $(Q_1, P_1), \ldots, (Q_s, P_s)$ as the primary decomposition of J.

Theorem 5.4. *Algorithm* PRIMARYDECOMPOSITIONALGEBRAIC *is correct.*

Proof. It is very easy to check that the several conditions imposed on the result are satisfied, using the basic homomorphic properties of ϕ. □

Remark 5.5.

(1) Kemper extends his radical computation algorithm from rational function fields to finitely presented algebraic function fields using exactly the same technique as here in Kemper (2002, Remark 8 (a)).
(2) In the MAGMA implementation, the user can define a function field by an arbitrary chain of algebraic and transcendental extensions in any order; given an ideal over this field, the code constructs the corresponding ideal over an equivalent field having the form $\mathbf{F}_q(t_1, \ldots, t_k)[\alpha_1, \ldots, \alpha_r]/I$, calls PRIMARYDECOMPOSITIONALGEBRAIC on this ideal, and finally converts the decomposition back to being over the original field. Thus very general presentations of fields are handled easily.

5.3. Positive-dimensional ideals

Suppose J is an ideal of dimension $d > 0$ defined over either type of function field explicitly described in the previous two subsections. Then we can now compute the

primary decomposition of J, applying a well-known method: we extend the base field by d new transcendental generators to form a zero-dimensional ideal and then can decompose this using one of the preceding zero-dimensional algorithms (which handle the transcendental generators), iterating as necessary (see Gianni et al. (1988, Section 8) or Becker and Weispfenning (1993, Section 8.7) for details).

This also covers in particular the surprisingly difficult problem of computing the primary decomposition of a positive-dimensional ideal of $\mathbf{F}_q[x_1, \ldots, x_n]$, for which there has apparently been no general algorithm described hitherto. (Kemper (2002, Introduction) points out that the radical of such an ideal can be computed with the same reduction method since his zero-dimensional algorithm handles transcendental generators.)

In theory, one could also use this algorithm to factor a polynomial f defined over the algebraic function field $K[\alpha_1, \ldots, \alpha_r]/I$ for maximal I by decomposing the principal ideal generated by f, but the following section presents a more direct and efficient algorithm for this (by computing a resultant).

In summary, in this section we have solved the problem of computing the primary decomposition of an ideal of any dimension defined over a rational function field or a general finitely presented algebraic function field (or, a fortiori, a finite field, covering the difficult positive-dimensional case).

6. Factorization

In this section, let K be the rational function field $\mathbf{F}_q(t_1, \ldots, t_k)$ of characteristic p and suppose that I is a maximal ideal of $K[\alpha_1, \ldots, \alpha_r]$, so $F_K = K[\alpha_1, \ldots, \alpha_r]/I$ is a field. We will present an algorithm for factoring general multivariate polynomials in $F_K[x_1, \ldots, x_n]$.

Lemma 6.1. *Let S be a separable radical of I over L, where L is some PIE of K. Let $F_L = L[\alpha_1, \ldots, \alpha_r]/S$. Then F_L is a field, and F_K is embedded naturally into F_L via $\phi : f + I \mapsto f + S$.*

Proof. S is maximal over L since it is the only prime component of I over L and is zero dimensional, so F_L is a field. Since $I \subseteq S$, ϕ is well defined and a homomorphism. Since ϕ is a non-zero homomorphism from a field to a field, ϕ is injective. □

Thus, given $F_K = K[\alpha_1, \ldots, \alpha_r]/I$, we can call SEPARABLERADICAL on I to obtain a separable radical S of I over some PIE L. (In the MAGMA implementation we usually need to call PRIMARYDECOMPOSITION(I) to prove that I is maximal to check that the user has defined a genuine field. When I is truly maximal, the single ideal S computed in Step 1 of PRIMARYDECOMPOSITION is kept and stored in I.)

Now since S is a separable radical, S possesses a shape basis generator by Theorem 4.5, so $F_L = L[\alpha_1, \ldots, \alpha_r]/S$ is isomorphic to the field $F_z = L[z]/\langle u(z)\rangle$, for some separable $u(z) \in L[z]$ (since S is separable). We now show that we can extend Trager's norm-based factorization algorithm (Trager, 1976) to a **separable** simple algebraic extension.

Theorem 6.2. *Suppose F_z is the field $L[z]/\langle u(z)\rangle$, where $u(z)$ is separable and irreducible over L, an infinite field. Let $g \in F_z[x_1, \ldots, x_n]$ be separable, and primitive in all variables. Write* $\mathrm{Norm}(h) = res_z(h, u)$.

1. *Let y_v be a variable such that the degree of g in y_v is non-zero. Then for all $s \in L$ except for a finite number of exceptions,* $\mathrm{Norm}(g_s)$ *is separable, where g_s is obtained from g by the substitution $y_v \mapsto y_v - sz$.*
2. *Suppose $N = \mathrm{Norm}(g_s)$ is separable. Let $h_1, \ldots h_l$ be the irreducible factors of N over L. Then*

$$g_s = \prod_{i=1}^{l} \mathrm{GCD}(g_s, h_i)$$

   is a complete factorization of g_s over F_z.
3. *g can be effectively factored over F_z into a product of irreducibles.*

Proof. Point 1 is Trager (1976, Theorem 2.3) or Geddes et al. (1992, Theorem 8.18), and Point 2 is Trager (1976, Theorem 2.2) or Geddes et al. (1992, Theorem 8.17), with 'squarefree' replaced with 'separable' in all cases, and the univariate case easily generalized to use the variable y_v; the proofs carry over with no other changes needed. Point 3 then follows: enumerate elements of L to find a good s (which must exist by Point 1 since L is infinite); perform the appropriate substitution; apply Point 2; and finally perform the inverse substitution on each irreducible factor. □

Thus to obtain a complete algorithm to factor a given $f \in F_K[x_1, \ldots, x_n]$, all we need do is move f to F_z, and compute the separable factorization of f, lying over some PIE F_{z_s} of F_z, and then we are able to apply Trager's algorithm to each separable factor, and compute each corresponding irreducible over the original field F_K. Here is the complete algorithm.

$$\begin{array}{ccc} & & F_{z_s} = L_s[z_s]/\langle d(z_s)\rangle \\ & & \uparrow \\ F_L = L[\alpha_1, \ldots, \alpha_r]/S & \cong & F_z = L[z]/\langle d(z)\rangle \\ \uparrow & & \\ F_K = K[\alpha_1, \ldots, \alpha_r]/I & & \end{array}$$

Algorithm FACTORIZATION(f)
INPUT: $f \in F_K[x_1, \ldots, x_n]$, where $F_K = K[\alpha_1, \ldots, \alpha_r]/I$, $K = \mathbf{F}_q(t_1, \ldots, t_k)$ of characteristic p, and I is a maximal ideal, so F_K is a field.
OUTPUT: The factorization of f over F_K.
STEPS:

1. Let $S \lhd L[\alpha_1, \ldots, \alpha_r]$ be a separable radical of I, for some PIE L of K (using algorithm SEPARABLERADICAL). Let $F_L = L[\alpha_1, \ldots, \alpha_r]/S$; F_K is embedded naturally in F_L.

2. Let s be a shape basis generator of S, with separable minimal polynomial $d(z)$ over L. Now $F_z = L[z]/\langle d(z)\rangle$ is isomorphic to F_L, and the shape basis of S gives the isomorphism explicitly.
3. Let $f_z \in F_z[x_1, \ldots, x_n]$ correspond to f. Apply algorithm SEPARABLEFACTORIZATION to f_z to obtain a separable factorization $\prod_{i=1}^{m} {g_i}^{e_i}$ over $F_{z_s} = L_s[z_s]/\langle d(z_s)\rangle$, where L_s is a PIE of L.
4. For each i, remove the contents in all variables from $g_i \in F_{z_s}$, and apply Trager's algorithm (via Theorem 6.2) to each primitive polynomial, to obtain factors $h_{i,j}$ for $j = 1, \ldots, l_i$, each of which is irreducible over F_{z_s}.
5. For each $h_{i,j}$, apply algorithm IDEALOVERSUBFIELD to F_K and the principal ideal of $F_{z_s}[x_1, \ldots, x_n]$ generated by $h_{i,j}$ to obtain a principal ideal whose generator is the corresponding irreducible $H_{i,j}$ over F_K; output each $H_{i,j}$ with multiplicity

$$e_i \cdot \frac{\text{TotalDegree}(h_{i,j})}{\text{TotalDegree}(H_{i,j})}.$$

Theorem 6.3. *Algorithm* FACTORIZATION *is correct.*

Proof. The previous discussion has shown the correctness of Steps 1–4. Now for each irreducible $h_{i,j} \in F_{z_s}[x_1, \ldots, x_n]$, Lemma 2.2(b) shows that for the minimal $l_{i,j}$ such that

$$H_{i,j} = h_{i,j}^{p^{l_{i,j}}} \in F_K[x_1, \ldots, x_n],$$

$H_{i,j}$ is irreducible over F_K. This also equals the generator of the principal ideal $\langle h_{i,j}\rangle \cap F_K[x_1, \ldots, x_n]$, and clearly

$$p^{l_{i,j}} = \frac{\text{TotalDegree}(H_{i,j})}{\text{TotalDegree}(h_{i,j})},$$

so the multiplicities are also correct. So the correct factorization of f into powers of irreducibles over F_K is returned. □

Remark 6.4.

(1) Note that only Steps 1 and 2 need be executed once for a fixed field F_K; then only Steps 3–5 need be executed for each input polynomial f over this F_K.
(2) As before, we do not need in practice to extend the fields occurring at any stage, since the only virtual extension occurs in Step 3 when we call SEPARABLEFACTORIZATION and this is applied to a polynomial defined over a field of the type covered in point (2) of Section 2.3.
(3) When IDEALOVERSUBFIELD is called in Step 5, the ideal J will contain $h_{i,j}$, a relation defining the generator z_s of F_{z_s} as a p^l-th root of the generator z of F_z, a relation defining z in terms of the generators of F_L and the defining relations of F_L (given by the shape basis of S), and defining relations for any roots of the original generators of F_K which are used in any of the above relations. After the monomial order is changed, $H_{i,j}$ will appear as the only polynomial involving the original generators of F_K and $x_1, \ldots, x_n$ alone. One can optimize this by calling IDEALOVERSUBFIELD(K, S) once and remembering the

full Gröbner basis B_2 of the ideal J in Step 2 of IDEALOVERSUBFIELD. This gives relations for the roots of the original generators in terms of the generators of F_L, and so the elimination step in FACTORIZATION can be made simpler because only the root z_s needs to be eliminated. (As noted above, the MAGMA implementation usually calls PRIMARYDECOMPOSITION(I) anyway to prove that the user has really defined a field, so IDEALOVERSUBFIELD(K, S) is already called there.)

(4) In Step 5, to compute the irreducible $H_{i,j}$ over F_K corresponding to the irreducible $h_{i,j}$ over F_{z_s}, one could instead power $h_{i,j}$ by successive powers of p until all coefficients of the polynomial are in F_K. However, to test this condition one would still need to perform an elimination computation similar to the optimized computation just mentioned (to eliminate z_s), so there is no advantage.

Example 6.5.

(1) Let F_K be the field $\mathbf{F}_2(t,u)[\alpha,\beta]/\langle\alpha^2+t, \beta^4+u\rangle$, and let

$$f = x^8 + (t+u)x^4 + tu \in F_K[x].$$

SEPARABLEFACTORIZATION(f) returns

$$f = g^4, \quad g = x^2 + (T+U)x + TU,$$

where $T = \sqrt[4]{t}$, $U = \sqrt[4]{u}$. The separable radical of the defining ideal of F_K is

$$\langle\alpha + T^2, \beta + U\rangle$$

which is already a shape basis. Trager's algorithm factors g as

$$g = (x+T)(x+U).$$

Applying IDEALOVERSUBFIELD to the corresponding principal ideals, we obtain

$$(x+T) \mapsto (x^2+\alpha) \text{ over } F_K,$$
$$(x+U) \mapsto (x+\beta) \text{ over } F_K.$$

So the final factorization over F_K is

$$f = (x^2+\alpha)^2(x+\beta)^4.$$

Note the differing multiplicities for the factors in the output, despite the fact that there was only one factor in the initial separable factorization (so this differs from factorization over perfect fields).

(2) Continuing Example 5.3(2), let $K = \mathbf{F}_2(t,u)$, and let $F_K = K[\alpha,\beta,\gamma]/I$, where

$$I = \langle\alpha^2 + t\beta,\ \beta^4 + \beta^2 + \gamma,\ \gamma^2 + tu\rangle.$$

Let us factor:

$$f = x^8 + t^2x^4 + t^4\gamma \in F_K[x].$$

SEPARABLEFACTORIZATION(f) yields

$$f = g^4, \text{ where } g = x^2 + \sqrt{t}x + t\sqrt[4]{\gamma}.$$

The previously given separable radical S of I can be written as

$$\langle \alpha + T^4\beta + T^5U,\ \beta^2 + \beta + T^2U^2,\ \gamma + T^4U^4 \rangle,$$

where $T = \sqrt[8]{t}$, $U = \sqrt[8]{u}$, and S has shape basis generator β. So setting

$$d = \beta^2 + \beta + T^2U^2,$$

we embed F_K in $\mathbf{F}_2(T, U)[\beta]/\langle d \rangle$ where $\gamma \mapsto T^4U^4$, so $\sqrt[4]{\gamma} \mapsto TU$. We thus obtain

$$g = x^2 + T^4x + T^8 \cdot TU,$$

and d and g are separable over $\mathbf{F}_2(T, U)$. Trager's algorithm then finds these two irreducible factors of g over $\mathbf{F}_2(T, U)[\beta]/\langle d \rangle$:

$$h_1 = x + T^4\beta + T^5U + T^4,$$
$$h_2 = x + T^4\beta + T^5U.$$

For h_1, we form the ideal J in $\mathbf{F}_2(t, u)[x, \alpha, \beta, \gamma, T, U]$ generated by

$$B_1 = \left\{ \begin{array}{c} x + T^4\beta + T^5U + T^4, \\ \alpha + T^4\beta + T^5U,\ \beta^2 + \beta + T^2U^2,\ \gamma + T^4U^4, \\ T^8 + t,\ U^8 + u \end{array} \right\}.$$

This is a lexicographical Gröbner basis with $x > \alpha > \beta > \gamma > T > U$. We then move to the lexicographical Gröbner basis of J with $T > U > x > \alpha > \beta > \gamma$ and obtain

$$B_2 = \left\{ \begin{array}{c} T + 1/(tu)U^3x\beta\gamma + 1/(tu)U^3\alpha\beta\gamma + 1/(tu)U^3\alpha\gamma, \\ U^4 + 1/tx\gamma + 1/t\alpha\gamma, \\ x^2 + t\beta + t, \\ \alpha^2 + t\beta,\ \beta^4 + \beta^2 + \gamma,\ \gamma^2 + tu \end{array} \right\}.$$

So $H_1 = x^2 + t\beta + t$ is the corresponding irreducible factor over F_K. Notice that the degree of H_1 is 2, so the multiplicity of H_1 in the output is $4/2 = 2$. Then h_2 is handled similarly, and this time the resulting factor is $H_2 = x + \alpha$ (which can in fact be seen from the relation defining α in S). This time the degree of H_2 is 1, so the multiplicity of H_2 in the output is $4/1 = 4$.
Thus the final factorization over F_K is

$$f = x^8 + t^2x^4 + t^4\gamma = (x + \alpha)^4(x^2 + t\beta + t)^2.$$

Notice again that f is a fourth power of a separable polynomial, but not a fourth power of a polynomial in $F_K[x]$.

7. Performance analysis

In this final section, we analyze the performance of the algorithms in practice. There are a number of issues to consider.

7.1. The simple algebraic extension

When we move to the simple algebraic extension in Step 2 of FACTORIZATION, the shape basis used for this transformation could conceivably be quite 'messy', and so expensive to compute. But since all these algorithms will be used in small characteristic in practice, there will be no blowup in the base coefficients in $\mathbf{F}_q$.

Of course, there could be some blowup in the coefficients (rational functions in the transcendental generators) of the defining polynomial of the simple algebraic extension. But this is inherent in the problem when we have a chain of relative extensions, and we have never found this to be a problem in practice for the algebraic fields with several algebraic generators which we have tried.

7.2. The GCD computations

The computation of polynomial GCDs may be very expensive (in the algorithm SEPARABLEFACTORIZATION, and when removing contents). Any GCD algorithms (such as the univariate Euclidean algorithm and the simple recursive multivariate subresultant algorithm (Knuth, 1998, Alg. 4.6.1.C)) would suffice for all of the algorithms in this paper to work correctly, but without a fast GCD algorithm, the algorithm SEPARABLEFACTORIZATION is hopeless for even moderately non-trivial examples! (The same problem arises when factoring over perfect fields.)

So the author has also developed and implemented an asymptotically fast evaluation–interpolation algorithm in MAGMA for computing the GCD of multivariate polynomials defined over an algebraic function field $\mathbf{F}_q(t_1, \ldots, t_k)(\alpha)$ with one algebraic generator. After reducing to the univariate polynomial case (by evaluation/interpolation), the algorithm basically works by evaluating and interpolating at each transcendental variable t_i; the base case involves computing GCDs in $(\mathbf{F}_q[\alpha]/\langle g(\alpha)\rangle)[x]$ (where g may be reducible), which is relatively fast. (The implementation requires about the same amount of code as the main algorithms of this paper!)

As in Step 2 of algorithm FACTORIZATION, we can move from an algebraic field having several algebraic generators to a field with only one generator (even if the transcendence degree is greater than one), so the asymptotically fast GCD algorithm needs to handle only simple algebraic extensions. The full Gröbner basis B_2 mentioned in Remark 6.4(3) allows one to map efficiently the GCD back to over the original field (with no eliminations needed, since the simple algebraic function field is not extended).

7.3. The Gröbner basis computations

The algorithm IDEALOVERSUBFIELD needs to compute a Gröbner basis. But we completely avoid using the Buchberger algorithm here, because it is easy to ensure that the monomial orders are such that the Gröbner Walk algorithm (Collart et al., 1997) can be used to compute all the Gröbner bases needed.

The FGLM algorithm (Faugère et al., 1993) could be used when IDEALOVERSUBFIELD is called in PRIMARYDECOMPOSITION, but is inadequate for Step 5 of FACTORIZATION, since the ideal constructed there will not be zero dimensional if there is more than one main polynomial variable. But we always use the Gröbner Walk because it works

very well even in the zero-dimensional case for the kinds of ideals constructed here: IDEALOVERSUBFIELD has usually taken negligible time for the examples we have tried.

7.4. The base multivariate factorizations

An arbitrary polynomial $f \in \mathbf{F}_q(t_1, \ldots, t_k)[x_1, \ldots, x_n]$ can be factored by an easy reduction to factorization over $\mathbf{F}_q$: simply clear the denominators of the coefficients of f to compute the corresponding $g \in \mathbf{F}_q[x_1, \ldots, x_n, t_1, \ldots, t_k]$, make g primitive in the variables $x_1, \ldots, x_n$, factor g, and then move each factor back to $\mathbf{F}_q(t_1, \ldots, t_k)[x_1, \ldots, x_n]$.

Now the base factorization of multivariate polynomials in $\mathbf{F}_q[y_1, \ldots, y_m]$ can suffer in standard algorithms from an exponential combinatorial problem.

However, a new algorithm (Belabas et al., submitted for publication) for factoring bivariate polynomials in $\mathbf{F}_q[x, y]$ has recently been developed, which uses ideas similar to those applied in van Hoeij's Knapsack factoring algorithm for $\mathbf{Z}[x]$ (van Hoeij, 2002). The new algorithm uses traces to gather linear relations which enable the correct combinations to be found in polynomial time (and very quickly in practice). On the basis of this algorithm, multivariate factorization can effectively be reduced to bivariate factorization (see Bernardin and Monagan (1997), for example) so that the combinatorial problem also does not arise in practice for more than two variables.

All these algorithms have been implemented in MAGMA, so the author's implementation of algorithm FACTORIZATION **never** encounters any exponential combinatorial problem.

7.5. Conclusion

All of the potentially expensive components of the algorithms presented here perform very well in practice in the MAGMA implementation.

Indeed, one can construct an inseparable algebraic function field of very small characteristic with a few transcendental generators and a few algebraic relations and then it usually takes less than a second on an average machine to factor a multivariate polynomial over the field which may take several screens to print, has many factors with various multiplicities, and may involve several non-trivial levels for the separable factors. (The GCD computations in SEPARABLEFACTORIZATION often take more than half the time.) See the web page (Steel, 2004) for examples.

So the algorithms presented are very successful in practice, and are also straightforward to implement, if one already has implementations of the standard characteristic-zero algorithms for primary decomposition and factorization over algebraic extensions.

Acknowledgments

This research was partially funded by the Defence Sciences and Technology Organization, Australia.

Thanks are expressed to the anonymous referees for several helpful comments.

References

Amrhein, B., Gloor, O., Kuechlin, W., 1996. Walking faster. In: Calmet, J., Limongelli, C. (Eds.), Design and Implementation of Symbolic Computation Systems. In: Springer LNCS, vol. 1128. pp. 150–161.

Becker, T., Weispfenning, V., 1993. Gröbner Bases. Springer Verlag, New York.

Belabas, K., van Hoeij, M., Klüners, J., Steel, A., 2004. Factoring polynomials over global fields. Int. Math. Res. Not. (submitted for publication).

Bernardin, L., Monagan, M., 1997. Efficient multivariate factorization over finite fields. In: Proc. AAECC'97. In: Springer LNCS, vol. 1255. pp. 15–28.

Bosma, W., Cannon, J., Playoust, C., 1997. The magma algebra system I: the user language. J. Symbolic Comput. 24 (3), 235–265. URL: http://magma.maths.usyd.edu.au.

Collart, S., Kalkbrener, M., Mall, D., 1997. Converting bases with the Gröbner walk. J. Symbolic Comput. 24 (3–4), 465–469.

Davenport, J.H., Trager, B.M., 1981. Factorization over finitely generated fields. In: Proceedings of the 1981 Symposium on Symbolic and Algebraic Computation. Snowbird, Utah, pp. 200–205.

Faugère, J.-C., Gianni, P., Lazard, D., Mora, T., 1993. Efficient computations of zero-dimensional Gröbner bases by change of ordering. J. Symbolic Comput. 16, 329–344.

Geddes, K., Czapor, S., Labahn, G., 1992. Algorithms for Computer Algebra. Kluwer, Boston.

Gianni, P., Trager, B.M., Zacharias, G., 1988. Gröbner bases and primary decomposition of polynomial ideals. J. Symbolic Comput. 6 (2–3), 149–167.

Kemper, G., 2002. The calculation of radical ideals in positive characteristic. J. Symbolic Comput. 34 (3), 229–238.

Knuth, D.E., 1998. The Art of Computer Programming, 3rd ed. vol. 2. Addison Wesley, Reading, MA.

Steel, A., 2004. Conquering inseparability (examples web page). http://magma.maths.usyd.edu.au/users/allan/insep.

Trager, B.M., 1976. Algebraic factoring and rational function integration. In: Jenks, R.D. (Ed.), Proc. SYMSAC'76. ACM Press, pp. 196–208.

van Hoeij, M., 2002. Factoring polynomials and the knapsack problem. J. Number Theory 95 (2), 167–189.